Adabas Audit Data Viewer

Table of contents

1 Introduction

Adabas Audit Data Viewer (AAV) is a web tool for auditors for viewing logs stored in connected databases. AAV allows to filter data (based on provided search criteria) and display details of found data.

AAV system consist of the following components:

- **Backend** - the part of AAV, which is not visible for the user of the web browser. Backend supports operation like login or searching data, based on search criteria provided via Web Client. The backend is a part of Mainframe environment, where Adabas database is available
- **Web Client** - frontend, the part of AAV, which is visible for the user of web browser, after the user enters AAV's web address.

2 Installation

Adabas Audit Data Viewer's installation package is provided as a WAR file which should be installed in a Java Application Server.

The tool was tested on web browsers: Chrome 72.0+, Firefox 60.0+, IE 77.0+, Safari 12.0+.

To ensure the proper and secure usage of AAV it is very important to follow also the installation and configuration instructions of chosen Java Application Server. The Java Application Server shall be installed, configured and operated only by qualified system administrator.

The configuration of secure TLS communication is recommended, but is not a part of this documentation. For that please refer to the instruction corresponding to chosen Java Application Server.

2.1 Installation requirements

The following components have to be installed:

- JDK (Java Development Kit) or JRE (Java Runtime Environment),
- Java Application Server.

2.2 Installation process

AAV installation package consist of two parts:

- WAR file that contains the AAV backend application, which must be deployed on Java Application Server,
- Web application, that contains static files and can be deployed on any web server (including Java Application Server).

2.3 Configuration process

2.3.1 AAV backend configuration

The backend configuration is placed in the application.yml file.

In application.yml file the administrator can define the following parameters:

- **BQL.SQL –** allows to enable SSL globally. Defines keystore and truststore paths and passwords. If it is set to true, the (absolute) paths to the keystore/truststore must be provided

```
 4    bql:
 5      ssl:
 6        enabled: true
 7    #    keystore:
 8    #      path:
 9    #      password:
10    #    truststore:
11    #      path:
12    #      password:
```

- **Endpoints**
    - **IP** – IP address, where AAR subsystem can be reached (range: 10.0.0.0 – 10.255.255.255),

- **PORT** – port number on the z/OS side where AAR subsystem can be reached (range 0 – 65535),
- **SSID** – AAR subsystem ID,
- **APPL** – application identifier (B97_AAV)
- **SSL** – defines, if the SSL connection shall be used. If this parameter is set to 'true', it has to be also enabled globally by means of the parameter 'bql/sql'. By default it is set to 'false'

```
31  server:
32
33    tomcat:
34      basedir: tomcat
35      accesslog:
36        directory: logs
37        enabled: true
38        pattern: common
39
40  endpoints:
41    Adabas_Data_Viewer_Demo_System:
42      ip:
43      port:
44      ssid:
45      aapl:
46    demo:
47      ip:
48      port:
49      ssid:
50      aapl:
51      ssl:
```

Remark: The port number (together with bind address and IP stack) is defined on the z/OS side via the B97_TCPIP_PORT_AAV parameter.

- **Client: allowed-origins**

The client allowed-origins defines the URLS which can connect to AAV's backend.

The following values can be defined for this parameter:

- list of allowed URLs,

```
13  client:
14      allowed-origins:"http://aav.test.com;http://aav.test2.com"
15
```

- all URLs allowed (can be used only for testing purposes).

```
13  client:
14      allowed-origins:"*"
15
```

- **User: inactivity**

    The time (in seconds) of user inactivity after which the user will be automatically logged out
    from the AAV.

```
22  user:
23      inactivity: 3600
24
```

- **Folder caching**

    This section of application.yml allows to improve the system performance by enabling Folder
    Tree caching mechanism:

    -   interval: the time after which the list of database folders is updated with newest data,

    -   enabled (true/false): enabling/disabling caching mechanism,

    -   limit: determines the maximum number of user sessions for which the cache
        mechanism is enabled, this parameter protects against overuse of a server resources.

        Example: If the number of users exceeds a set value, then the last logged-in users
        take over the resources of the first ones.

```
30  folder:
31      caching:
32          interval: 300
33          enabled: true
34          limit: 50
35
```

2.3.2 AAV frontend configuration

Depending on how the frontend files are served, the configuration must be changed accordingly.
Frontend configuration is available in config.js file in the frontend directory. This file contains
information about the AAV backend (WAR) URL and at which base path the frontend is served on.
These values are from the user's perspective, because of this URLs and paths configured here are the
public accessible URLs, as typed in browser address bar, not local file-system paths.

2.3.2.1 Setting API endpoint

API_URL defines the path or URL at which the API server is accessible. URL can be relative if the backend is hosted on the same domain or absolute if server is hosted on another domain. When AAV backend server is served from another domain, ensure that CORS is properly configured so that it can be accessed.
For example, if frontend is available at yourserver.com/aav and backend server is
on the same domain at yourserver.com/aav-api, then set API_URL variable as below:

```
window.config = {
 // ...
 API_URL: '/aav-api',
 // ...
};
```

If backend is on a different domain, set API_URL to a full URL, as in:

```
window.config = {
 // ...
 API_URL: '//api.yourserver.com/aav',
 // ...
};
```

2.3.2.2 Setting Web App base path

BASE_PATH defines path at which the AAV webapp is available on your domain. By default, it assumes it is available at "/", the root directory. If it is installed in a subdirectory, the configuration must be changed as described below.

If it is available on yourserver.com/aav, then set this value to /aav:

```
window.config = {
 // ...
 BASE_PATH: '/aav',
 // ...
};
```

Additionally, index.html file must also be updated with new path.
Replace <base href="/" /> with:

```
<base href="/aav/" />
```

The path must end with slash character /.

2.3.3 Routing configuration

AAV web app uses browser-side routing (html5 history). For this to work, your web server must serve index.html at paths that don't exist. AAV web app comes with WEB-INF directory, that redirects any invalid paths to index.html, which should work with most Java Application Servers.

2.4 Example of minimal configuration in Tomcat

1. If not installed, install **JDK (Java Development Kit)** or **JRE (Java Runtime Environment)**.
2. If not installed, install **Tomcat v.8.0+**.

    Links:

    - JDK installation: https://www.oracle.com/java/technologies/javase-jdk14-downloads.html
    - Tomcat installation: https://tomcat.apache.org/download-90.cgi

    Remark: During installation process, please ensure that the versions of Java and Tomcat (32-bit or 64-bit) will correspond to the operating system installed on the target computer.

3. Put files from AAV installation package to the Tomcat root folder (path_to_tomcat/).

    Remark: The AAV installation package for Tomcat contains the additional file **context.patch.** This file contains the information about the updates which should be done in **context.xml** file. The updates in context.xml file can be done manually (described in next step) or automatically (execution of the **context.patch** file).

4. Enable/disable the usage of the SSL connection by means of the BQL.SQL parameter.
5. Define endpoints:
    - update conf/context.xml file with a path to application.yml file:

    ```
    <Context>
        ...
        <Parameter name="spring.config.location" value="path_to_application.yml" />
        ...
    </Context>
    ```

    - define endpoints in conf/application.yml file.

```
31  server:
32
33    tomcat:
34      basedir: tomcat
35      accesslog:
36        directory: logs
37        enabled: true
38        pattern: common
39
40  endpoints:
41    Adabas_Data_Viewer_Demo_System:
42      ip:
43      port:
44      ssid:
45      aapl:
46    demo:
47      ip:
48      port:
49      ssid:
50      aapl:
51      ssl:
```

IP – IP address, where AAR subsystem can be reached (range: 10.0.0.0 – 10.255.255.255)

PORT – port number on the z/OS side where AAR subsystem can be reached (range: 0 - 65535)

SSID – AAR subsystem ID

APPL – application identifier (B97_AAV)

SSL – defines, if the SSL connection shall be used. If this parameter is set to 'true', it has to be also enabled globally by means of the parameter 'bql/sql'. By default it is set to 'false'

Remark: The port number (together with bind address and IP stack) is defined on the z/OS side via the B97_TCPIP_PORT_AAV parameter.

6. Configure in conf/application.yml file a client allowed-origins.

The client allowed-origins defines the URLs which can connect to AAV's backend.

It should be defined as follows:

- list of allowed URLs,

```
13  client:
14      allowed-origins:"http://aav.test.com;http://aav.test2.com"
15
```

- all URLs allowed (only for testing purposes).

```
13   client:
14       allowed-origins:"*"
15
```

7. Configure in conf/application.yml file a user session inactivity parameter.

The time (in seconds) of user inactivity after which the user will be automatically logged out from the AAV.

```
22   user:
23       inactivity: 3600
24
```

8. Configure in conf/application.yml a folder caching parameter

This section of conf/application.yml allows to improve the system performance by enabling Folder Tree caching mechanism:

- interval: the time after which list of database folders is updated with newest data,
- enabled (true/false): enabling/disabling caching mechanism,
- limit: determines the maximum number of user sessions for which the cache mechanism is enabled, this parameter protects against overuse of a server resources.

  Example: If the number of users exceeds a set value, then the last logged-in users take over the resources of the first ones.

```
30   folder:
31     caching:
32         interval: 300
33         enabled: true
34         limit: 50
35
```

9. Start (if Tomcat was installed in step 2) or restart Tomcat.
10. Configure AAV web application config.js. Depending on the release contents:
    - if the release package webapps directory contains page directory and app.war file (pre 1.0):

      Set API_URL to /app and BASE_PATH to /page in config.js:

      window.config = {
       API_URL: '/app',
       BASE_PATH: '/page',
      };

      And replace <base href="/"/> in index.html with:

<base href="/page/" />

- If the release package webapps directory contains aav directory and aav-api.war file (starting v1.0):

  Set API_URL to /aav-api and BASE_PATH to /aav in config.js:

  ```
  window.config = {
   API_URL: '/aav-api',
   BASE_PATH: '/aav',
  };
  ```

  And replace <base href="/"/> in index.html with:

  <base href="/aav/" />

11. To configure the secured TLS connection please refer to the following instruction: https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html

3 Login and change password

Login
To run AAV:

1. Enter the address of AAV in one of supported web browsers.
2. AAV presents login screen:

3. Fill the login form with the following data:

    a. Endpoint,

    b. Username (RACF* user account),

    c. Password

    and press **Login** button.

   *RACF (Resource Access Control Facility)** - the software which main features are related to access control management, like for example: identification and verification of the users by

means of their RACF login and corresponding to its password, maintenance of access rights to protected resources, identification and management of protected resources.

4. If the login data was correct, then the application presents Main screen, where the user can trigger further actions (described in next chapters).
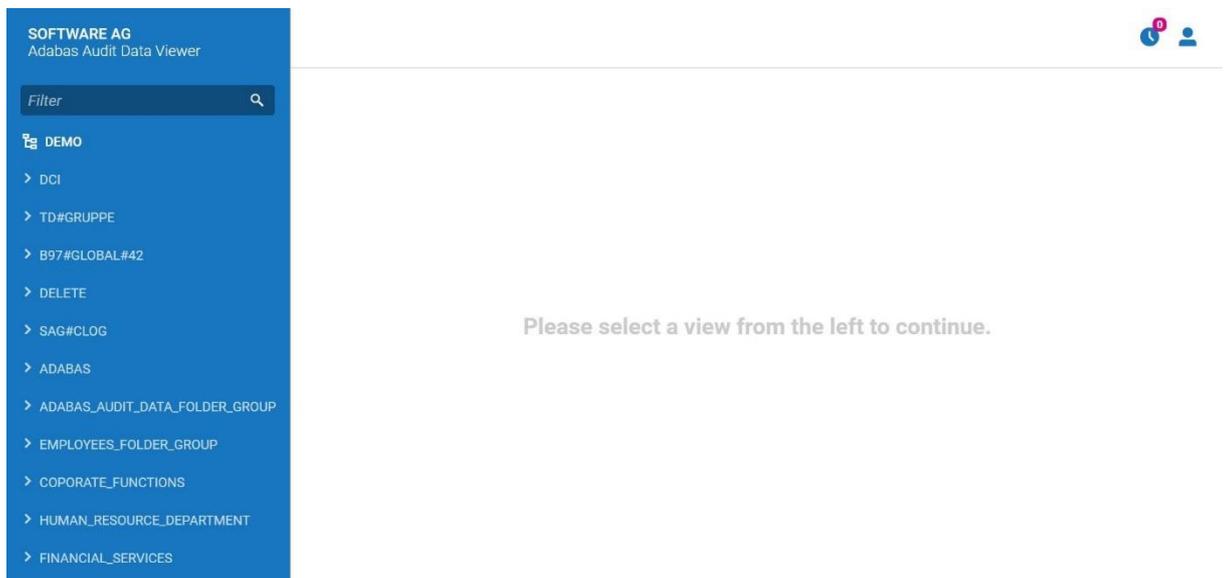
5. If the login data was not correct, then the application will present corresponding error screen:

Error: Your username and/or password are invalid   ✕

Log in to your account

Endpoint

demo ▼

Username

USER1

Password

··· 👁

☑ Remember me   Change password

Login

6. If the login attempt took place from a different network than server is running, then AAV will present corresponding error screen:

Change password

The user can change its password, by pressing the corresponding button.

Log in to your account

Endpoint

demo ▼

Username

USER1

Password

👁

☑ Remember me   ⇨ Change password

Login

After pressing the **Change password** button, AAV asks for the following information:

- old password,
- new password
- confirmation of the new password.

To save the changes, the user shall press **Change** button.

< Change password

Endpoint

demo ▼

Username

USER1

Old password

👁

New password

👁

Confirm new password

👁

Change

4 User Profile

1. On the top right corner of the Main screen there is the User Profile button located.

SOFTWARE AG
Adabas Audit Data Viewer

Filter 🔍

DEMO

> DCI

2. After pressing a User Profile Button, AAV presents available options:
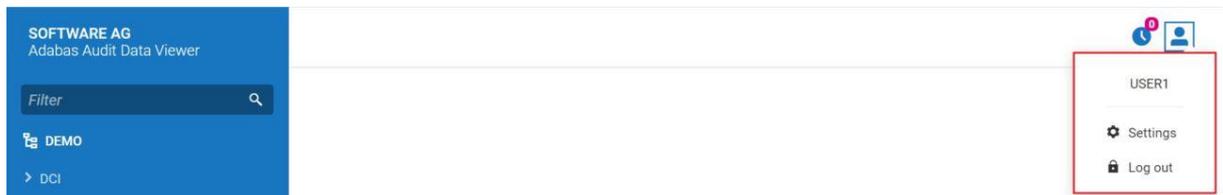
- the username of the currently logged user,
- settings, where the user can change:
  - pooling time, which defines interval in seconds at which the scheduled reports are updated,
  - date format - the format of date used in search forms and for results, changing this setting may re-run the current query,
  - list size limit - the limit of lists shown in search result,
  - setting related to showing or hiding all greyed-out fields (the fields which cannot be filled).

all the changes are saved after pressing **Save** button,

- log out, where the user can log out from AAV.

5 Layouts

1. On the left side of the Main screen there are layouts organized as a tree elements folder structure. The list and contents of each layout depends on databases connected to AAV.
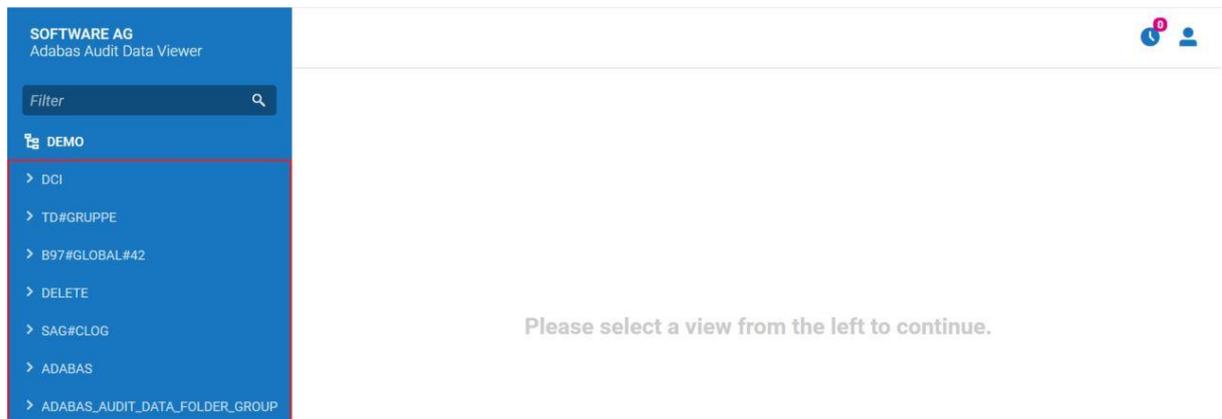
2. To find the layout from the list, the user can use one of the following options:
    - choose the layout from the list,
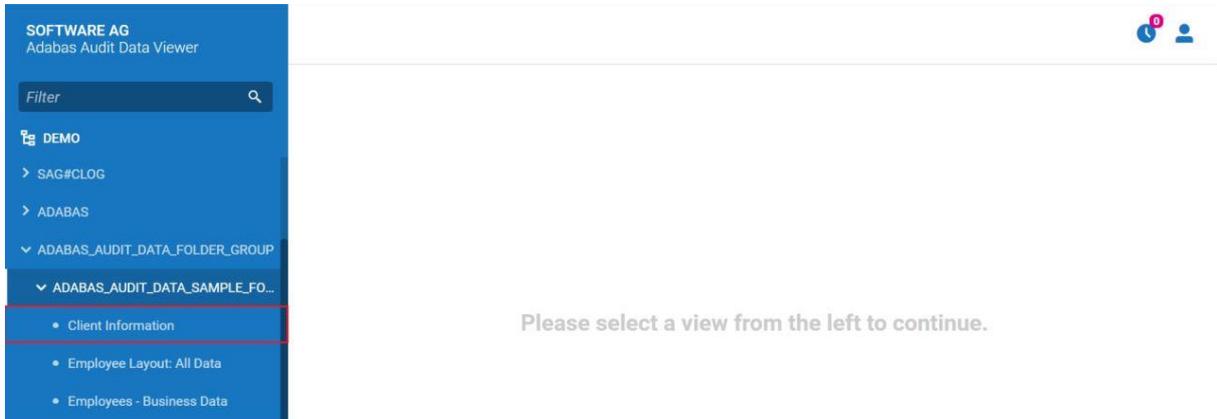
- by means of the filter field.
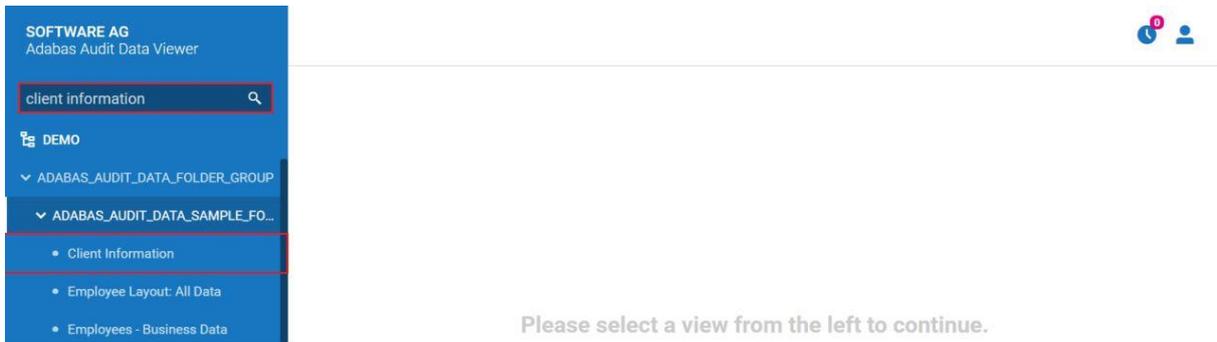
3. The user clicks on the chosen layout.
4. AAV presents the screen, where the user can add search criteria.

6 Logs Searching

The searching of the logs is done based on chosen layout and by adding additional search criteria. The available search criteria can be different for different layouts and depends on connected database.

To search the logs the following steps shall be done:

1. Choose the layout from the list.

2. AAV displays the screen, which allows to add search criteria (different for different layouts).
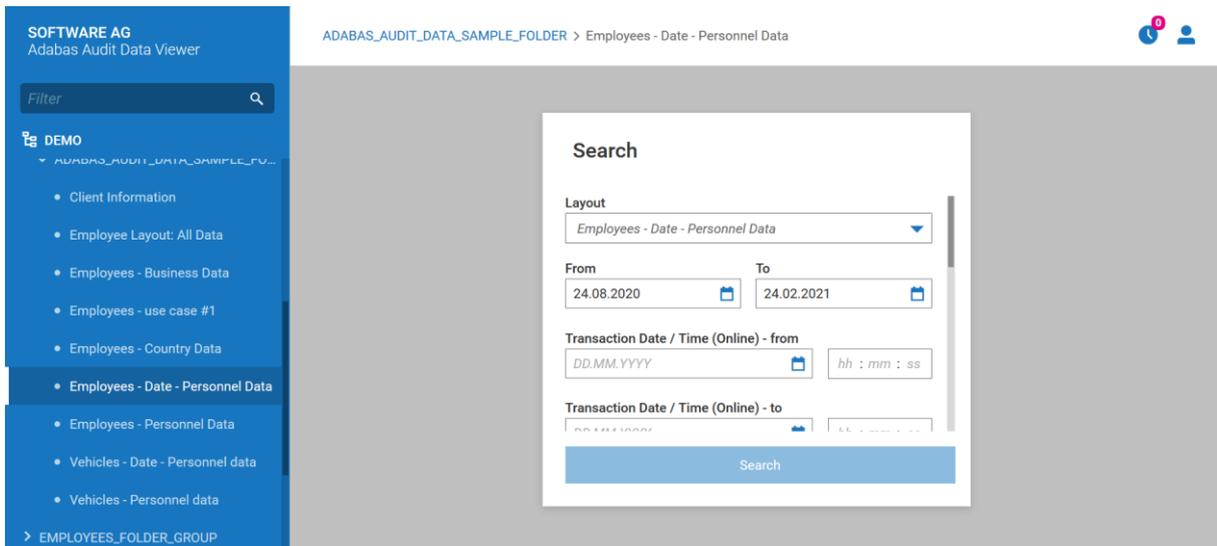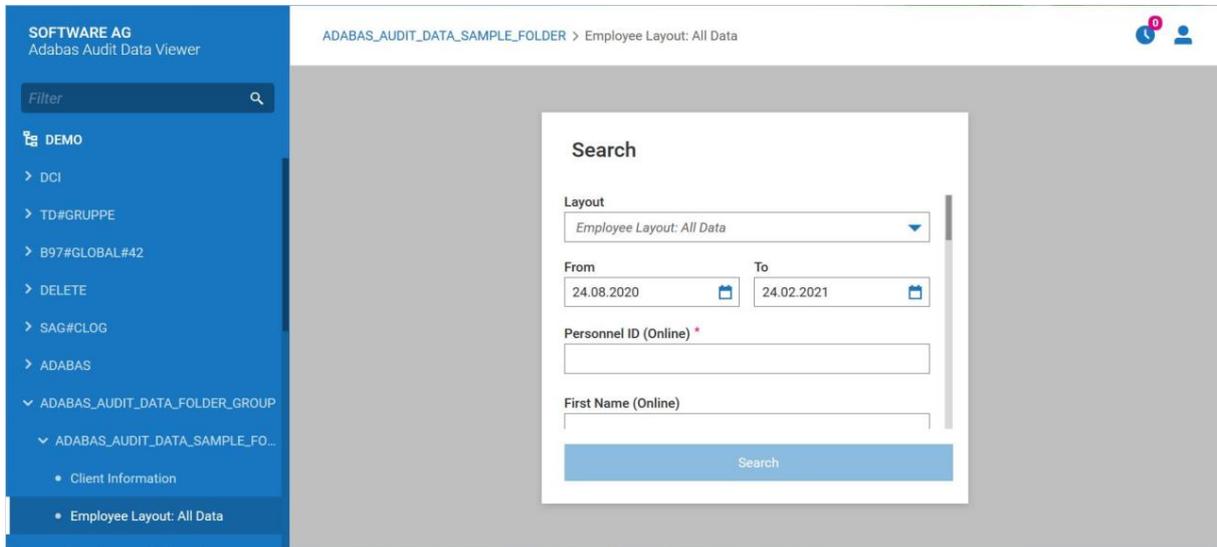
The user can change the layout using one of the following methods:

- by choosing layout from menu,
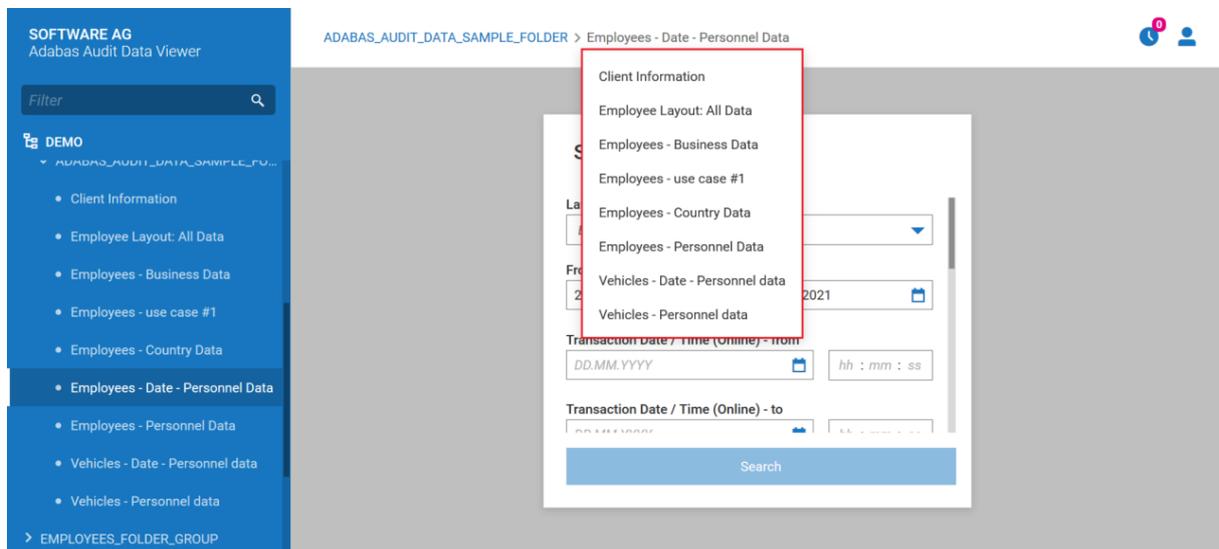
- by choosing layout from Search Criteria screen.
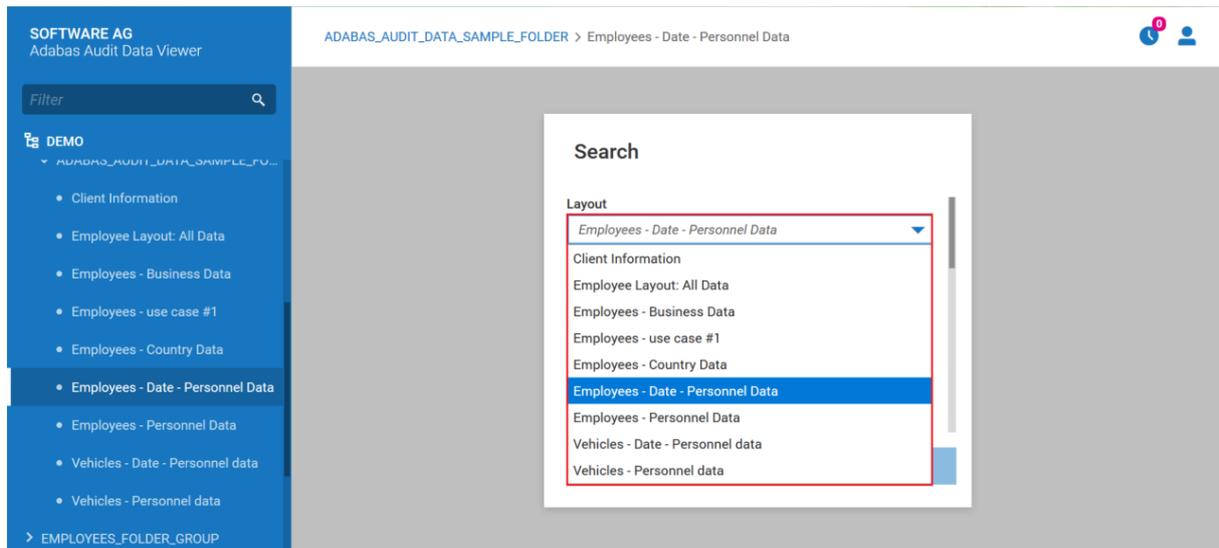
Remarks:

- AAV does not allow to execute the searching without adding at least one additional criteria, besides Date From/Date To,

- The mandatory fields are marked with "*",

- In Personnel ID field it is not allowed to add the following characters: ",'.

3. Add search criteria and press **Search** button.
4. AAV presents the search result:

- by choosing **Refresh** button, the user can reload the search result, for example after update of List size limit (function described in chapter 4 User Profile):

- by pressing **Settings** button, the user can define the columns (and their names), which will be presented in search result and row height:

- by adding text in Search field, the user can find lists, which match the text:

- by pressing **Edit search** button, the user can change search criteria (see point 2) and change search criteria:

- if there is no data for search result for provided search criteria, then AAV presents the message:

5. The user can refine the search result by choosing the list(s), adding additional search criteria and pressing **Refine Search** button.

6. AAV presents screen, where the user can add more search criteria.

At this step it is possible to start new search. It can be done by changing the layout.

If the user marks the option **Hide all disabled (greyed out) fields** and:

- one Audit File was chosen (it can be a multiple selection of one file as in example below), then AAV presents only fields which can be filled,

- different Audit Files (for example EMPLADA and VEHIADA) were chosen, then only Personnel ID field will be presented.

7. After adding additional search criteria, the user shall press **Search** button.
8. AAV presents refined search result.

9. To view the details of given list from refined search result, the user shall click on this list.

10. AAV presents the details of chosen list, where the user can:
    - view the items, by choosing them from the list:

After the first chosen item was presented, the user can also navigate between items from the list using **Prev** and **Next** buttons:

- check the details of each item corresponding to a given list:

In this view the following Tabs may be presented:

- Read Request
- Data Before Image
- Data After Image
- Client Info
- Request Info

- download the report which contains search result related to given list:
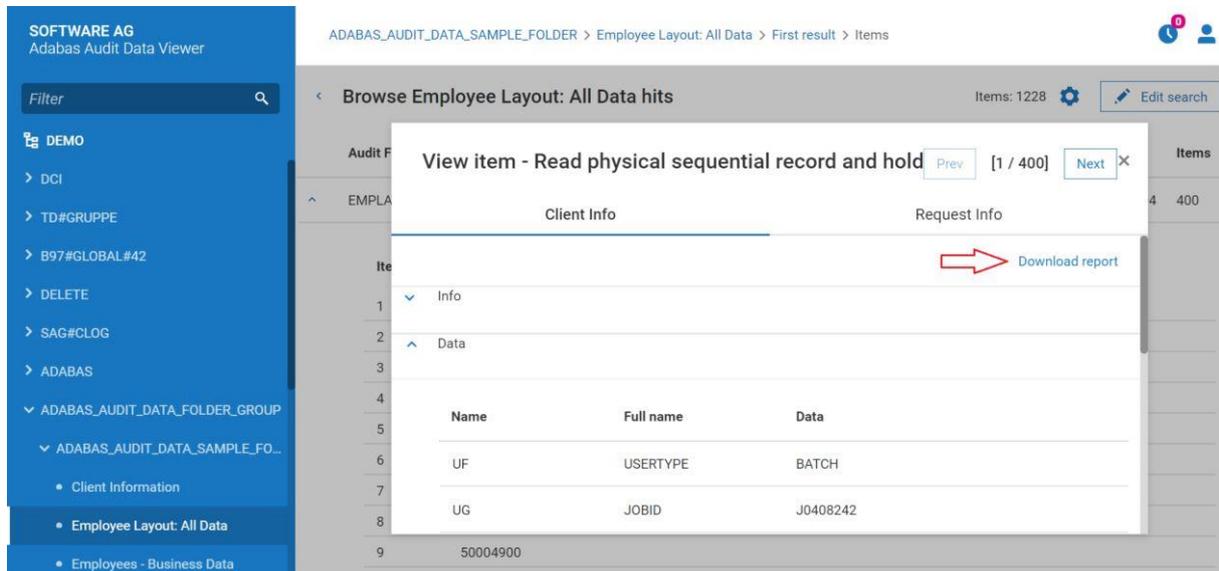
The report is downloaded as a binary file. The name of the file is defined based on schema AAV_report_etoken_currentDate_currentTime.alog.

The previewing of binary file is not supported by AAV.

7 Reports Scheduling

The **scheduled reports** option allows to schedule creation of the report for offline data (data, which are not available immediately, which are stored for example on a tape).

If one or more of the fields on search form have attribute **Offline** and user clicks on **Search** button, then AAV will ask the user which operation should be executed:

      a. wait for search result (this process may take several minutes),
      b. schedule the report, by adding report name and pressing **Schedule** button.

If the user chose option B, then, after pooling time, the report will be available in Scheduled reports and the user can display the details of the report.

When the report is ready, then the icon marked with an arrow becomes green.

The AAV does not delete the report, after it was opened by the user, this can be done manually.

The user can execute the following actions:

- view the report by clicking on the report from the list,

- delete a single report from the list by clicking "x",

- delete all the reports by clicking bin icon.

8 Error messages

The error messages are used to inform the user about problems which occurred during the use of the application.

The errors messages related to the database are presented with an error code from database.

AAV supports the following error messages:

1. while login process:
    - 'Your username and/or password are invalid' - appears when during login process not correct password or username was entered,
    - 'Could not connect to server' - appears in case of the connection attempt from different network than server's network,
    - 'An unknown error occurred during login (xxx)' - appears when the server responded to AAV with one of the following error codes: 400 or 500. Error message in AAV contains a description of an error from the server (instead of 'xxx'),
    - 'Server not reachable' - appears when AAV could not reach the server. This can happen when one of the following occurred: server is out of reach; server is turned off
2. while using the application:
    - 'Server not reachable' - appears when AAV could not reach the server. This can happen when one of the following occurred: server is out of reach, server is turned off, the connection with server's network was lost,
    - 'Could not load the folders' - appears when AAV was not able to load folders into tree element structure,
    - 'Could not load the schema for the view' - appears when during global search (described in chapter Logs Searching points 2 and 3), AAV did not received schema corresponding to chosen view,
    - 'Could not load the schema' - appears when during local search (described in chapter Logs Searching points 6 and 7), AAV did not receive schema corresponding to chosen view,
    - 'Could not load data' - appears when the AAV did not receive from the server data, which correspond to search criteria provided by the user,
    - 'Could not load user's settings' - appears when the AAV did not receive from the server information about settings of currently logged user,
    - 'Could not save user's settings' - appears when AAV was not able to save user settings due to server error,
    - 'Could not load status of scheduled reports' - appears when AAV was not able to load statuses of scheduled reports from the server,
    - 'Index format is inconsistent. A Search is not possible on all selected lists. Uncheck list causing the trouble and retry operation' – appears due to data format inconsistency.

9 Third Party Software

This section provides a list of the third-party software that is included in Adabas Audit Data Viewer together with the license terms under which this third-party software is distributed.

Backend Licenses

Software	License
Jakarta Annotations API (1.3.5)	EPL 2.0 GPL w/CPE
Jakarta Expression Language 3.0 (3.0.3)	EPL 2.0 GPL w/CPE
Logback Classic Module (1.2.3)	Eclipse Public License – v.1.0 GNU Lesser General Public License
Logback Core Module (1.2.3)	Eclipse Public License – v.1.0 GNU Lesser General Public License
Apache Commons Codec (1.14)	Apache License v.2.0
Apache Commons Lang (3.10)	Apache License v.2.0
Apache Log4j API (2.13.3)	Apache License v.2.0
Apache Log4j to SLF4J Adapter (2.13.3)	Apache License v.2.0
Bean Validation API (2.0.1.Final)	Apache License v.2.0
Byte Buddy (without dependencies) (1.10.11)	Apache License v.2.0
ClassMate (1.5.1)	Apache License v.2.0
Generex (1.0.2)	Apache License v.2.0
Guava: Google Core Libraries for Java (30.0)	Apache License v.2.0
Hibernate Validator Engine (6.1.5.Final)	Apache License v.2.0
Jackson datatype: jdk8 (2.11.0)	Apache License v.2.0

Jackson datatype: JSR310 (2.11.0)	Apache License v.2.0
Jackson-annotations (2.11.0)	Apache License v.2.0
Jackson-core (2.11.0)	Apache License v.2.0
jackson-databind (2.11.0)	Apache License v.2.0
Jackson-module-parameter-names (2.11.0)	Apache License v.2.0
Jakarta Bean Validation API (2.0.2)	Apache License v.2.0
Java Faker (1.0.2)	Apache License v.2.0
JBoss Logging 3 (3.4.1.Final)	Apache License v.2.0
MapStruct Core (1.2.0.Final)	Apache License v.2.0
SnakeYAML (1.26)	Apache License v.2.0
spring-security-config (5.3.3.RELEASE)	Apache License v.2.0
spring-security-core (5.3.3.RELEASE)	Apache License v.2.0
spring-security-web (5.3.3.RELEASE)	Apache License v.2.0
springfox-core (2.9.2)	Apache License v.2.0
springfox-schema (2.9.2)	Apache License v.2.0
springfox-spi (2.9.2)	Apache License v.2.0
springfox-spring-web (2.9.2)	Apache License v.2.0
springfox-swagger-common (2.9.2)	Apache License v.2.0
springfox-swagger-ui (2.9.2)	Apache License v.2.0
springfox-swagger2 (2.9.2)	Apache License v.2.0
swagger-annotations (1.5.20)	Apache License v.2.0

swagger-models (1.5.20)	Apache License v.2.0
tomcat-embed-core (9.0.36)	Apache License v.2.0
tomcat-embed-websocket (9.0.36)	Apache License v.2.0
Automaton (1.11-8)	BSD
Spring AOP (5.2.7.RELEASE)	Apache License v.2.0
Spring Beans (5.2.7.RELEASE)	Apache License v.2.0
Spring Commons Logging Bridge (5.2.7.RELEASE)	Apache License v.2.0
Spring Context (5.2.7.RELEASE)	Apache License v.2.0
Spring Core (5.2.7.RELEASE)	Apache License v.2.0
Spring Expression Language (SpEL) (5.2.7.RELEASE)	Apache License v.2.0
Spring Web (5.2.7.RELEASE)	Apache License v.2.0
Spring Web MVC (5.2.7.RELEASE)	Apache License v.2.0
spring-boot (2.3.1.RELEASE)	Apache License v.2.0
spring-boot-autoconfigure (2.3.1.RELEASE)	Apache License v.2.0
spring-boot-starter (2.3.1.RELEASE)	Apache License v.2.0
spring-boot-starter-json (2.3.1.RELEASE)	Apache License v.2.0
spring-boot-starter-logging (2.3.1.RELEASE)	Apache License v.2.0
spring-boot-starter-security (2.3.1.RELEASE)	Apache License v.2.0
spring-boot-starter-tomcat (2.3.1.RELEASE)	Apache License v.2.0
spring-boot-starter-validation (2.3.1.RELEASE)	Apache License v.2.0

Software	License
spring-boot-starter-web (2.3.1.RELEASE)	Apache License v.2.0
Spring Plugin - Core (1.2.0.RELEASE)	Apache License v.2.0
Spring Plugin - Metadata Extension (1.2.0.RELEASE)	Apache License v.2.0
JUL to SLF4J bridge (1.7.30)	MIT
SLF4J API Module (1.7.30)	MIT
java jwt (3.10.3)	MIT

Frontend Licenses

Software	License
@babel/preset-env (7.10.4)	MIT
@fortawesome/fontawesome-free (5.13.0)	CC-BY-4.0 AND OFL-1.1 AND MIT
@testing-library/jest-dom (4.2.4)	MIT
@testing-library/react (9.5.0)	MIT
@testing-library/user-event (7.2.1)	MIT
@types/dotenv (8.2.0)	MIT
@types/events (3.0.0)	MIT
@types/jest (24.9.1)	MIT
@types/lodash (4.14.157)	MIT

@types/node (14.0.1)	MIT
@types/react (16.9.35)	MIT
@types/react-datepicker (3.0.2)	MIT
@types/react-dom (16.9.8)	MIT
@types/uuid (8.3.0)	MIT
dotenv (8.2.0)	BSD-2-Clause
events (3.2.0)	MIT
lodash (4.17.15)	MIT
mobx (5.15.4)	MIT
mobx-react (6.2.2)	MIT
moment (2.25.3)	MIT
query-string (6.13.1)	MIT
react (16.13.1)	MIT
react-app-polyfill (1.0.6)	MIT
react-datepicker (3.0.0)	MIT
react-dnd (11.1.3)	MIT
react-dnd-html5-backend (11.1.3)	MIT
react-dom (16.13.1)	MIT
react-router (5.2.0)	MIT
react-router-dom (4.3.1)	MIT
react-scripts (3.4.1)	MIT

typescript (3.7.5)	Apache-2.0
uuid (8.3.0)	MIT
rollup/plugin-eslint (8.0.0)	MIT
rollup-plugin-includepaths (0.2.4)	MIT
classnames (2.2.6)	MIT
apidevtools/swagger-cli (4.0.4)	MIT
css-loader (2.1.0)	MIT
style-loader (0.23.1)	MIT
swagger-ui (3.20.7)	Apache-2.0
yaml-loader (0.5.0)	MIT

10 Change history

Version	Date	Author	Reviewed by	Approved by	Change history
0.1	30.09.2020	K.Filipiuk			Document ready for review.
0.2	09.10.2020	K.Filipiuk			Rework after review: review comments were implemented, screens in chapters 6 and 7 were updated.
0.3	11.12.2020	K.Filipiuk			Documentation update: description about Login Window - change password possibility, description about Download file functionality, List of Third Party Software updated, information about Folder Caching mechanism, Info about User session inactivity feature added, screenshots reflect the current App Layout added.
0.2.1	21.01.2021	K.Filipiuk			Documentation update: installation chapter updated.
1.0.0.	27.01.2021	K.Filipiuk			Documentation ready for review.